Re-parametrising cost matrices for tuning model predictive controllers

Chin, Robert; Rowe, Jonathan E.

DOI:
10.1109/CEC.2019.8790260

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Chin, R & Rowe, JE 2019, Re-parametrising cost matrices for tuning model predictive controllers. in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC 2019).* IEEE Computer Society Press, pp. 2856-2863, 2019 IEEE Congress on Evolutionary Computation (CEC 2019), Wellington, New Zealand, 10/06/19. https://doi.org/10.1109/CEC.2019.8790260

Re-parametrising Cost Matrices for Tuning Model Predictive Controllers

Robert Chin
Department of Electrical & Electronic Engineering
The University of Melbourne
Melbourne, Australia
and
School of Computer Science
University of Birmingham
Birmingham, United Kingdom
robert.chin@unimelb.edu.au

Jonathan E. Rowe
School of Computer Science
University of Birmingham
Birmingham, United Kingdom
and
The Alan Turing Institute
British Library
London, United Kingdom
J.E.Rowe@cs.bham.ac.uk

***Abstract*—In control systems engineering, the selection of controller parameters play an important role in obtaining optimal controller performance. However, it is often not possible to obtain closed-form relationships between the parameters and performance, making the selection process difficult. This paper presents an automated tuning strategy for Model Predictive Controllers (MPC) whereby a meta-cost function is introduced to penalise undesirable behaviour, and subsequently optimised over using black-box search algorithms. To this end, we propose a method of re-parametrising the cost matrices in MPC. This approach results in a box-constrained parametrisation for the matrices, as well as a reduction in the search dimension. The procedure is demonstrated on a diesel engine case study, where we compare the tuning of MPC using the proposed parametrisation to an unbounded parametrisation on a test suite of optimisation algorithms: Simulated Annealing (SA), Particle Swarm Optimisation (PSO), Genetic Algorithms (GA), Nesterov's gradient-free algorithm (NGF) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). We find that the proposed parametrisation provides a statistically significant advantage on all algorithms tested except CMA-ES, for which the performance was similar. We discuss this latter empirical result in relation to the theoretical invariance properties of CMA-ES.**

***Index Terms*—diesel engines, evolutionary computation, optimization, predictive control**

I. INTRODUCTION

Model Predictive Control (MPC) [1] is a suitable strategy for controlling constrained multiple-input multiple-output (MIMO) dynamical systems because of its ability to explicitly handle state and input constraints while solving for an optimal input sequence over a finite time-horizon. Traditionally, MPC has been favoured in the process industry, where the slower dynamics and sampling times of systems allowed for the MPC online optimisation problem to be solved in between samples. However, increases in the availability of computing power have allowed for MPC to be implemented on systems with faster dynamics and sampling times, such as automotive diesel engines [2], [3].

The first author is supported by the Elizabeth and Vernon Puzey Scholarship (The University of Melbourne) and the Priestley Scholarship (The University of Melbourne & University of Birmingham).

A problem faced in the calibration of MPC are the tuning of the cost matrices that appear in the cost functional, namely the Q, P, R matrices in a quadratic formulation with terminal cost. While the Q, P, R matrices are intended to penalise deviations of the states and inputs over an open-loop prediction horizon, the impact of each element in these matrices on higher-level objectives (say, time-domain characteristics) of controller performance with closed-loop feedback is not obvious. Experienced control engineers (such as automotive engine calibrators) have gained an innate understanding of time-domain characteristics desirable in system trajectories. However, the unintuitive relationship between Q, P, R and time-domain characteristics leads to an increased burden associated with tuning and commissioning new controllers, and acts as a barrier for MPC to be more commonplace in industrial domains, including automotive applications. The issue of tuning MPC has been widely addressed in literature, although not comprehensively solved. Several textbooks [1], [4] provide heuristic guidelines for practitioners to tune MPC cost parameters. In special cases (i.e. when constraints are not active and when the matrices are of simpler structure), analytical results have yielded explicit tuning rules for parameters [5]. In [6], an envelope-constrained formulation of MPC was developed to be more amenable for tuning by engineers.

In offline tuning of MPC (in which cost parameters are tuned via simulation before being tested in practice), several authors have approached the tuning problem by formulating a higher-level objective function of the closed-loop MPC response, which we will refer to in this paper as a "meta-cost" function. These authors have then used an optimisation algorithm in conjunction with simulations of the closed-loop MPC to optimise the meta-cost with respect to the parameters of the MPC [7], [8].

In this paper, we build on the state-of-the-art for offline tuning of MPC via a meta-cost in the following way. We present a method for re-parametrising the Q, P, R matrices which reduces the search dimension while retaining the full positive-definite structure, and also transforming the problem to be over a box-constrained search space. Some previous literature [9], [10] have only considered tuning MPC with simplified cost structure, such as when Q, R are constrained to be effectively diagonal or scaled identity matrices. Investigating more freedom in the structure of the matrices allows the potential for better controllers to be found. In addition to algorithms used in existing MPC tuning literature, we propose the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) optimisation algorithm for solving the meta-cost tuning problem. We show that our tuning approach is also compatible with existing stability guarantees of linear, constrained systems. Lastly, we present results on an automotive diesel engine case study to benchmark this parametrisation in comparison with another parametrisation on a test suite of optimisation algorithms.

II. PROBLEM STATEMENT

A. Notation

The set of real numbers is denoted $\mathbb{R}$, the set of positive real numbers is denoted $\mathbb{R}_{>0}$, and the set of natural numbers is denoted $\mathbb{N}$. The symbol $\top$ denotes the matrix transpose. The superscript $+$ denotes a forward shift in discrete-time. Inequalities involving vectors are to be interpreted element-wise. Matrix-valued variables will be capitalised and the symbol $\succ$ is reserved for matrix inequalities.

B. General Formulation

A general formulation of the controller tuning problem may be posed as follows. Suppose that a controller is parametrised by $\vartheta \in \Theta$, and there exists a real-valued function $\mathcal{J}(\vartheta) : \Theta \to \mathbb{R}$ that maps each parameter value to a scalar performance index. Then we consider the tuning problem as to be finding ideally:

$$\vartheta^* = \underset{\vartheta \in \Theta}{\operatorname{argmin}}\, \mathcal{J}(\vartheta) \tag{1}$$

C. MPC Formulation

In this paper, we concentrate on the problem of tuning MPC matrices of a quadratic cost functional with terminal cost for regulation of a linear system with linear state and input constraints. The linear discrete-time model is a dynamical system with inputs u, outputs y, and states x given by a difference equation of the form:

$$x^+ = Ax + Bu \tag{2}$$

and output equation:

$$y = Cx \tag{3}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$ with linear state constraints $Mx \le f$, linear input constraints $Eu \le h$ and a slew rate constraint $|\Delta u| := |u^+ - u| \le \widetilde{u}$. Consider a control objective of choosing inputs to regulate the system to the origin $x = 0$ from an initial condition x_0. At sampling instant k, the online MPC cost function is defined as a quadratic in the states and inputs with prediction horizon $N \in \mathbb{N}$:

$$V_k = \sum_{i=0}^{N-1} \left(x_{k|i}^\top Q x_{k|i} + u_{k|i}^\top R u_{k|i} \right) + x_{k|N}^\top P x_{k|N} \tag{4}$$

where $x_{k|i}$ denotes the predicted future state at time $k+i$ from current state x_k, and $u_{k|i}$ denotes the applied input at time $k+i$. Also, $Q \in \mathbb{R}^{n\times n}$, $P \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are symmetric positive-definite weighting matrices to penalise the deviation of future states and inputs from the origin. We take the prediction horizon N to be given (because in practice this could be dependent on hardware limitations), and thus is not considered as a tuning parameter. At each sampling instant k, the controller finds the optimal open-loop sequence $\{u_k^*\} := \left\{ u_{k|0}^*, \ldots, u_{k|N-1}^* \right\}$ by solving:

$$\begin{aligned} \min_{u_{k|0},\ldots,u_{k|N-1}} \quad & V_k \\ \text{subject to } \quad & x_{k|i+1} = Ax_{k|i} + Bu_{k|i},\ i = 0,\ldots,N-1 \\ & Mx_{k|i} \le f,\ i = 1,\ldots,N \\ & Eu_{k|i} \le h,\ i = 0,\ldots,N-1 \\ & \left| u_{k|i} - u_{k|i-1} \right| \le \widetilde{u},\ i = 0,\ldots,N-1 \end{aligned} \tag{5}$$

and applies $u_{k|0}^*$ as the control law at time k. Hence the closed-loop sequence of controls for times $k = 0, 1, 2, \ldots$ will be given by $\left\{ u_{0|0}^*, u_{1|0}^*, u_{2|0}^*, \ldots \right\}$. This online optimisation problem can be formulated as a quadratic program (QP), hence it is a convex problem and there exist a variety of QP solvers that can aid in finding the global optimum.

D. Meta-cost Formulation

We formulate the offline MPC tuning problem using a meta-cost function. Suppose that in a finite $T \in \mathbb{N}$ number of steps, the closed-loop dynamics of the MPC with given initial condition x_0, closed-loop control sequence $\left\{ u_{0|0}^*, \ldots, u_{T-1|0}^* \right\}$ and cost matrix tuple $\{Q, P, R\}$ yields the following sequence of outputs:

$$\mathcal{Y}_{\{Q,P,R\}} := \left\{ y_{0|\{Q,P,R\}}, \ldots, y_{T|\{Q,P,R\}} \right\} \tag{6}$$

Denote $J\left(\mathcal{Y}_{\{Q,P,R\}}\right)$ as a meta-cost function of the closed-loop output response. Then, the meta-cost tuning problem for MPC is formulated as:

$$\begin{aligned} \min_{\{Q,P,R\}} \quad & J\left(\mathcal{Y}_{\{Q,P,R\}}\right) \\ \text{subject to } \quad & Q, P, R \succ 0 \end{aligned} \tag{7}$$

where the decision variables exist in continuous space. Since the meta-cost can be an arbitrary function of the closed-loop response, it is neither guaranteed to be convex nor smooth, hence the tuning problem should be treated as a black-box optimisation problem, where we may attempt to apply heuristic search algorithms to find sub-optimal solutions.

III. RE-PARAMETRISATION

In this section, a method for re-parametrising the tuple $\{Q, P, R\}$ is presented. This re-parametrisation is motivated by the observation that if we multiply Q, P, R all by the same positive constant $c > 0$, the solution to the MPC problem (5) remains unchanged. This suggests that there is some redundancy by naively parametrising $\{Q, P, R\}$ by the elements of the matrices.

Additionally, we perform dimension reduction by fixing the terminal cost weight P with respect to Q and R. Suppose there is an implicit function $g(P,Q,R)$ of the cost matrices. We choose a scheme where given some Q and R, then P is fixed to be the matrix which solves $g(P,Q,R)=0$. In proceeding, we further require $g(P,Q,R)$ to satisfy the following assumption.

Assumption 1: If the tuple $\{Q_1,P_1,R_1\}$ solves $g(P_1,Q_1,R_1)=0$, then for any $c>0$, then the tuple $\{cQ_1,cP_1,cR_1\}$ also solves $g(cP_1,cQ_1,cR_1)=0$. Furthermore, if $Q_1\succ 0$ and $R_1\succ 0$, then $P_1\succ 0$.

Hence this allows us to concentrate only on parametrising Q and R. An important consideration is that Q and R must be symmetric positive-definite, which allows us to parametrise the matrices with only their upper or lower triangular elements. Although a perturbation to these values (such as that applied by a search algorithm) does not guarantee the matrices will remain positive-definite, it is possible to enforce positive-definiteness by projection onto the positive-definite cone (as addressed in [7]), by using the spectral decomposition (ie. eigendecomposition) of positive-definite matrices. The method presented in this paper is also based on eigendecomposition, from which we may write:

$$\begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} = \begin{bmatrix} W_Q & 0 \\ 0 & W_R \end{bmatrix} \begin{bmatrix} D_Q & 0 \\ 0 & D_R \end{bmatrix} \begin{bmatrix} W_Q^\top & 0 \\ 0 & W_R^\top \end{bmatrix} \quad (8)$$

where W_Q, W_R are orthogonal matrices and D_Q, D_R are diagonal matrices. The orthogonal matrix W_Q may be minimally parametrised by $\left(n^2-n\right)/2$ parameters (see [11] for a survey covering various approaches to parametrise orthogonal matrices), and likewise W_R may be minimally parametrised by $\left(m^2-m\right)/2$ parameters. In this paper, we choose the method of parametrisation to be the Givens rotations, which are a generalisation of the Euler rotations. This is also a method of minimal parametrisation, so we can choose $\left(n^2-n\right)/2$ angles in $[-\pi,\pi)$ to parametrise W_Q and $\left(m^2-m\right)/2$ angles in $[-\pi,\pi)$ to parametrise W_R.

The diagonal matrix $\operatorname{diag}\{D_Q,D_R\}$ is parametrised using $n+m-1$ parameters, if we impose the constraint that $\operatorname{trace}\left(D_Q^2\right)+\operatorname{trace}\left(D_R^2\right)=1$. Under this scheme, we are effectively constraining the diagonals to be a point on the unit hypersphere in $n+m$ dimensions, which is a $n+m-1$-dimensional manifold. Thus using hyperspherical coordinates, we can parametrise both D_Q and D_R using $n+m-1$ angles in $(0,\pi/2)$ so that we a picking from the positive orthant $\mathbb{R}^{n+m-1}_{>0}$, hence all the diagonals will be positive to satisfy the positive-definite requirement of Q and R. Although in choosing this parametrisation we have lost the ability to choose the scale of the matrices (ie. radius of the hypersphere), this is of no consequence as it has been established that scaling Q, P, R by some positive constant does not change the performance of the controller. In choosing this parametrisation, we have also removed the redundancy in parameters as originally alluded to.

We take stock to count the total number of parameters under this re-parametrisation. We have $\left(n^2+n\right)/2+\left(m^2+m\right)/2-1$ parameters, compared to $\left(n^2+n\right)/2+\left(m^2+m\right)/2$ if Q and R were parametrised by their elements. Hence the number of parameters has been reduced by 1, but we have also enforced the positive-definite requirement on Q and R over a box-constrained parametrisation. Compared to fully parametrising Q, P, R by their elements, we have reduced the number of dimensions by $\left(n^2+n\right)/2+1$, but lost the flexibility of choosing P independently of Q and R. However, we argue that this is not of great detriment. The rationale of P as a finite-horizon terminal cost weighting is to approximate the cost weighting over an infinite horizon from N to ∞. If there are no constraints, P has a solution which gives the exact weighting over an infinite horizon. This solution is the solution to the discrete-time algebraic Riccati equation [12]:

$$Q+A^\top PA-A^\top PB\left(R+B^\top PB\right)^{-1}B^\top PA-P=0 \quad (9)$$

Note that taking $g(P,Q,R)$ to be the discrete-time algebraic Riccati equation satisfies Assumption 1. In Section V, it is also described how this choice of P can guarantee stability. A simple alternative option which also satisfies Assumption 1 is to fix $P=Q$. With these examples we show that there are satisfactory approaches for fixing P, in lieu of choosing P independently of Q and R.

A. Re-parametrisation with a Regularisation Factor

We also provide an alternative re-parametrisation technique for situations where a practitioner wishes to regularise the solution to the MPC. If it is known that the uncontrolled system $x^+=Ax$ is asymptotically stable at the origin (ie. the spectral norm $\|A\|<1$), then setting $u_k=0$ for all k will ensure the system autonomously regulates to the origin in the absence of disturbances. In this case, the role of MPC may typically be to provide feedback and drive the states to the origin quicker than would be by setting $u_k=0$ for all k. However if there is plant-model mismatch, that is, if the prediction model used by the MPC is not the same as the actual dynamics of the system, then the MPC will be applying inputs based on ill-informed predictions. In these cases, there is merit to weighting deviations of the input above deviations of the states because then the input solved by MPC would be 'closer' to the known valid input $u_k=0$. Thus the matrix R relative to Q and P represents a form of regularisation on the solution to the online MPC problem.

Introduce a parameter $\rho>0$. In this alternative parametrisation, the orthogonal matrices W_Q and W_R are parametrised as before, however now D_Q and D_R are separately parametrised with hyperspherical coordinates such that $\operatorname{trace}\left(D_Q^2\right)=1$ and $\operatorname{trace}\left(D_R^2\right)=\rho^2$. Thus there are now $\left(n^2+n\right)/2+\left(m^2+m\right)/2-2$ parameters to be optimised and $\rho>0$ is chosen by the practitioner based on their confidence in the predictive model, with a higher model confidence leading to reduced ρ.

IV. OPTIMISATION ALGORITHMS

The method in Section III for parametrising the tuple $\{Q,P,R\}$ allows the meta-cost tuning problem to be solved over a box-constrained search space. While this makes it convenient to write routines for simple black-box optimisation algorithms such as grid search and random search to solve (7), in this paper we consider the use of more sophisticated solvers including the CMA-ES algorithm [13], particle swarm optimisation (PSO) [14], genetic algorithms (GA) [15], and Nestorov's gradient-free method (NGF) [16]. With the exception of CMA-ES, each of the aforementioned class of search strategies has been previously employed in the tuning/design of MPC ([17] for PSO, [18] for GA and [7] for NGF). For the sake of comparison, we also consider the adaptive simulated annealing (SA) algorithm [19].

CMA-ES operates on a statistical principle by representing the distribution of possible candidate solutions by a Gaussian distribution parametrised by a mean and covariance. At each iteration, a sample of candidate solutions is evaluated on the objective ('fitness') function, and the relative fitnesses are used to update the mean and covariance in a maximum-likelihood fashion. CMA-ES has theoretical foundations in information geometry [20] and close links with natural gradient descent, which utilises the Fisher information matrix to update the probability distribution between iterations. It is also shown that a particular variant of the CMA-ES algorithm can be recovered as a time-discretisation and finite-sample approximation of the Information-Geometric Optimisation (IGO) flow in continuous time, when applied to the Gaussian family of distributions [21]. CMA-ES has been experimentally demonstrated to benchmark well against other derivative-free optimisation algorithms on test problems over a continuous space which are particularly rugged and ill-conditioned [22]–[24]. On this encouraging note, it stands to reason that we further investigate CMA-ES in a practical application to MPC tuning.

A. Benefits of Re-parametrisation

We remark on potential benefits of using the box-constrained re-parametrisation in optimisation. While by some intuitive notion, a re-parametrisation which transforms an unbounded space into a bounded space might appear to 'shrink' the search space, it is not immediately clear or not whether this will prove beneficial. To illustrate with a counter-example, consider the unbounded search space $\Omega = \{\omega \in \mathbb{R} : \omega > 1\}$ with decision variable ω, and the re-parametrisation $\omega' = 1/\omega$ so that the re-parametrised search space is given by $\Omega' = \{\omega' \in \mathbb{R} : 0 < \omega' < 1\}$. While the search space has been transformed to now be box-constrained on the interval $(0, 1)$, an algorithm may struggle to find good local optima if the re-parametrisation concentrates all the local optima close to 0.

However in our case, we have exploited the redundancy in the Q, P, R matrices to simultaneously reduce the search dimension while re-parametrising onto a bounded space. For each $\{Q, P, R\}$ naively parametrised by its elements, there exists a local search direction (which is proportional to $\{Q, P, R\}$ itself), whereby a step along this direction will not affect the meta-cost value, as $\{Q, P, R\}$ will have only been modified by a scaling. Sacrificing rigour for exposition, an algorithm which does not explicitly consider derivatives may 'waste' function evaluations by searching in (or close to) this direction, while a non-population-based algorithm may be 'tricked' into thinking it has found a local optimum at that point. Re-parametrising to the box-constrained search space will remove this redundant search direction, possibly leading to increased efficiency in optimisation.

Furthermore, another benefit of this box-constrained re-parametrisation is that the decision variables will be guaranteed to stay within known bounds, avoiding numerical computation issues (such as ill-conditioning or rounding errors) related to the decision variables growing unbounded during search.

V. Stability Guarantees

In this section, we summarise how the meta-cost tuning method can be made to be compatible with existing results of closed-loop stability guarantees in MPC, for scenarios where stability guarantees are required. A simple example illustrating closed-loop stability is provided in Appendix A. For systems where human-safety is important and/or instability leads to catastrophic consequences (as can found in numerous automotive and aerospace examples), such guarantees are important. For simplicity, we state the conditions required for stability of deterministic, linear, constrained systems with meta-cost tuning. A wider survey of results in stability of MPC are found in [25], [26] and can be adapted in a similar manner.

Assumption 2: The system dynamics and MPC predictive model are both given by (2).

Assumption 3: The polytopes for state constraints $\mathbb{X} := \{x : Mx \leq f\}$ and input constraints $\mathbb{U} := \{u : Eu \leq h\}$ contain the origin.

Definition 1 (Maximum output admissible set): Let K^* denote the gain in the discrete-time Linear Quadratic Regulator (LQR) control law with dynamics (2) and cost weight matrices Q, R. The maximum output admissible set $\mathbb{X}_a$ is defined as:

$$\mathbb{X}_a := \{x : (A + BK^*)\, x \in \mathbb{X}, K^*\, (A + BK^*)\, x \in \mathbb{U}\} \quad (10)$$

An algorithm on computing $\mathbb{X}_a$ may be found in [27]. Introduce the region of attraction $\mathbb{X}_r$ as the set of states that can be steered by admissible control sequences (ie. satisfying state, input, and slew rate constraints) to $\mathbb{X}_a$ in N steps or less.

Assumption 4: The initial condition $x_0 \in \mathbb{X}_r$.

We then modify the MPC formulation with a terminal constraint as the maximum output admissible set:

$$\begin{aligned}
\min_{u_{k|0},\ldots,u_{k|N-1}} \; & \sum_{i=0}^{N-1} \left(x_{k|i}^\top Q x_{k|i} + u_{k|i}^\top R u_{k|i} \right) + x_{k|N}^\top P^* x_{k|N} \\
\text{subject to } & x_{k|i+1} = Ax_{k|i} + Bu_{k|i},\ i = 0, \ldots, N-1 \\
& x_{k|i} \in \mathbb{X},\ i = 1, \ldots, N-1 \\
& x_{k|N} \in \mathbb{X}_a \\
& u_{k|i} \in \mathbb{U},\ i = 0, \ldots, N-1 \\
& \left| u_{k|i} - u_{k|i-1} \right| \leq \widetilde{u},\ i = 0, \ldots, N-1
\end{aligned} \quad (11)$$

where P^* is fixed to be the solution of the discrete-time algebraic Riccati equation (9).

Theorem 1: Under Assumptions 1, 2, 3, 4, the closed-loop system using any MPC with cost matrices $\{Q, P, R\}$ with augmented formulation (11) found through meta-cost tuning (7) with the re-parametrisation approach in Section III is exponentially stable with region of attraction $\mathbb{X}_r$.

This guarantees that any candidate solution of the meta-cost tuning problem (7) using the re-parametrisation technique will be a stabilising controller.

VI. CASE STUDY

We demonstrate the meta-cost tuning approach for diesel engine air-path control. The schematic of the diesel engine air-path with exhaust gas recirculation (EGR) and variable geometry turbine (VGT) is displayed in Figure 1.

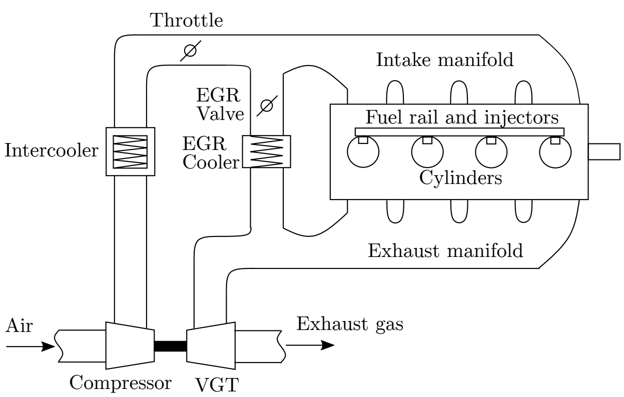

Fig. 1. A schematic of the a diesel engine air-path with exhaust gas recirculation (EGR) and variable geometry turbine (VGT).

A reduced 4th-order model structure developed in [3] has $n = 4$ states:

$$x = \begin{bmatrix} p_{\mathrm{im}} & p_{\mathrm{em}} & W_{\mathrm{comp}} & y_{\mathrm{EGR}} \end{bmatrix}^\top \tag{12}$$

where p_{im} is the intake manifold (boost) pressure, p_{em} is the exhaust manifold pressure, W_{comp} is the compressor mass flow rate and y_{EGR} is the EGR rate (which is the ratio of EGR mass flow rate to the sum of EGR and compressor mass flow rates). There are $m = 3$ actuators:

$$u = \begin{bmatrix} u_{\mathrm{thr}} & u_{\mathrm{EGR}} & u_{\mathrm{VGT}} \end{bmatrix}^\top \tag{13}$$

where u_{thr} is the throttle valve, u_{EGR} is the EGR valve and u_{VGT} is the VGT vane. The outputs of interest are the boost pressure and EGR rate, so that $y = \begin{bmatrix} p_{\mathrm{im}} & y_{\mathrm{EGR}} \end{bmatrix}^\top$. The dynamical model is parametrised by an engine speed N_{e} and fuel rate w_{fuel}, and the pair $(N_{\mathrm{e}}, w_{\mathrm{fuel}})$ as known as an operating point. For each engine operating point, there exists an input setpoint u_{ss} stored in the memory of the engine control unit (ECU), which induces an associated steady state x_{ss} as the steady state of the system when input u_{ss} is held fixed. For a given operating point $(N_{\mathrm{e}}, w_{\mathrm{fuel}})$, we refer to $(u_{\mathrm{ss}}, x_{\mathrm{ss}})$ as the operating condition at that operating point. Using experimental data from a test diesel engine at one particular operating point $(N_{\mathrm{e}}^*, w_{\mathrm{fuel}}^*)$, we identified using least-squares a local linear model with (A, B) trimmed about the operating condition $(x_{\mathrm{ss}}^*, u_{\mathrm{ss}}^*)$. The system is constrained by upper and lower bounds for states, saturation constraints for the inputs, and slew rate constraints.

The results of the system model identification for the local linear model at $(N_{\mathrm{e}}^*, w_{\mathrm{fuel}}^*)$ appear in Figure 2. The identification data was obtained by perturbing the system about the steady-state operating condition with a sum-of-sinusoids input signal. An open-loop simulated trajectory of the states (ie. given only the initial condition and sequence of inputs) using the identified model is compared against the obtained data to indicate the model fit. Diesel-engine dynamics are known for being highly nonlinear, sometimes requiring an 8th-order nonlinear model to faithfully describe [28]. Using a 4th-order local linear model, Figure 2 shows that the dynamics are still reasonably well-captured about the operating condition $(x_{\mathrm{ss}}^*, u_{\mathrm{ss}}^*)$. Although performing simulations with a higher-order nonlinear model is likely to lead to more accurate tuning results, for the purposes of demonstration and computational efficiency, we proceed in this paper with using the 4th-order local linear model to generate simulations.

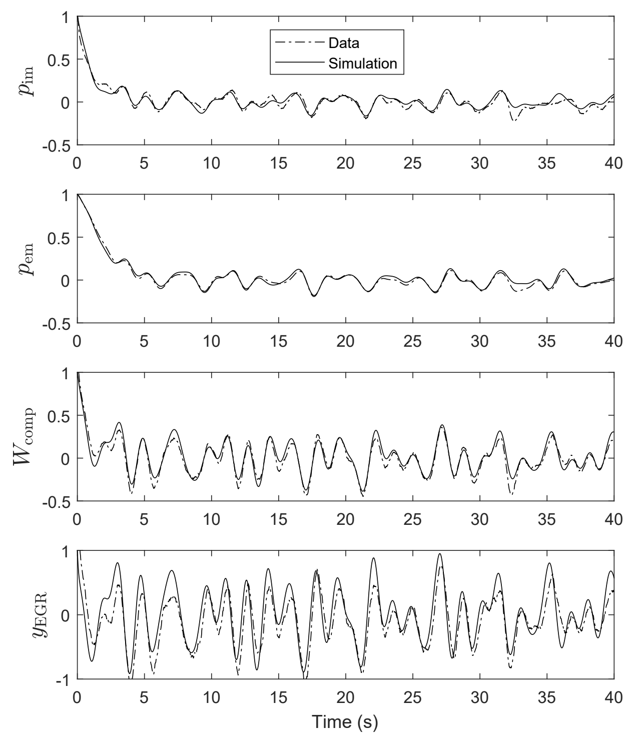

Fig. 2. System identification results for the 4th-order local linear model of the diesel engine air-path. The vertical axes are in normalised units.

We tune MPC to regulate the diesel engine at the operating condition for $(N_{\mathrm{e}}^*, w_{\mathrm{fuel}}^*)$. To parametrise the Q and R matrices, the 15-length vector is introduced:

$$\begin{aligned}&(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\phi})\\ &= (\theta_1, \theta_2, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \phi_1, \phi_2, \phi_3)\end{aligned} \tag{14}$$

with the box-constraints $-\pi \leq \boldsymbol{\theta} < \pi$, $-\pi \leq \boldsymbol{\phi} < \pi$ and $0 \leq \boldsymbol{\alpha} < \pi/2$. Using $s\theta := \sin\theta$ and $c\theta := \cos\theta$ as shorthand, we parametrise Q and R firstly with the Givens rotations:

$$
\begin{aligned}
W_Q = &\begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_2 & 0 & -s\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ s\theta_2 & 0 & c\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\times \begin{bmatrix} c\theta_3 & 0 & 0 & -s\theta_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s\theta_3 & 0 & 0 & c\theta_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_4 & -s\theta_4 & 0 \\ 0 & s\theta_4 & c\theta_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_5 & 0 & -s\theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & s\theta_5 & 0 & c\theta_5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c\theta_6 & -s\theta_6 \\ 0 & 0 & s\theta_6 & c\theta_6 \end{bmatrix} \qquad (15)
\end{aligned}
$$

$$
\begin{aligned}
W_R = &\begin{bmatrix} c\phi_1 & -s\phi_1 & 0 \\ s\phi_1 & c\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\phi_2 & 0 & -s\phi_2 \\ 0 & 1 & 0 \\ s\phi_2 & 0 & c\phi_2 \end{bmatrix} \\
&\times \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi_3 & -s\phi_3 \\ 0 & s\phi_3 & c\phi_3 \end{bmatrix} \qquad (16)
\end{aligned}
$$

and then hyperspherical coordinates:

$$D_Q = \operatorname{diag}\left\{ \begin{bmatrix} c\alpha_1 \\ s\alpha_1 c\alpha_2 \\ s\alpha_1 s\alpha_2 c\alpha_3 \\ s\alpha_1 s\alpha_2 s\alpha_3 c\alpha_4 \end{bmatrix} \right\} \qquad (17)$$

$$D_R = \operatorname{diag}\left\{ \begin{bmatrix} s\alpha_1 s\alpha_2 s\alpha_3 s\alpha_4 c\alpha_5 \\ s\alpha_1 s\alpha_2 s\alpha_3 s\alpha_4 s\alpha_5 c\alpha_6 \\ s\alpha_1 s\alpha_2 s\alpha_3 s\alpha_4 s\alpha_5 s\alpha_6 \end{bmatrix} \right\} \qquad (18)$$

to form $Q = W_Q D_Q W_Q^\top$ and $R = W_R D_R W_R^\top$. Then P is fixed with respect to Q and R by solving the discrete-time algebraic Riccati equation in (9) (for which MATLAB provides a routine with the `dare` command).

For this demonstration, we design a simple meta-cost function from a convex combination of time-domain characteristics of the output response. The meta-cost function is expressed by:

$$
\begin{aligned}
J\left(\mathcal{Y}_{\{Q,R,P\}}\right) = &\ \sigma_1 \mathtt{RT}_1 + \sigma_2 \mathtt{ST}_1 + \sigma_3 \mathtt{OS}_1 + \sigma_4 \mathtt{US}_1 \\
&+ \sigma_5 \mathtt{RT}_2 + \sigma_6 \mathtt{ST}_2 + \sigma_7 \mathtt{OS}_2 + \sigma_8 \mathtt{US}_2 \qquad (19)
\end{aligned}
$$

with $\sum_{i=1}^{8} \sigma_i = 1$, where each characteristic is defined in Table I.

In this particular case study we choose $\sigma_1 = 0.3$, $\sigma_2 = 0.01$, $\sigma_3 = 0.02$, $\sigma_4 = 0.01$, $\sigma_5 = 0.1$, σ_6, $\sigma_7 = 0.45$, $\sigma_8 = 0.01$. MPC is implemented with the QPKWIK algorithm [29] as the QP solver in closed-loop simulation with the identified linear model, using the formulation (5). Each simulation is initialised with the operating condition of a nearby operating point and run for 5 seconds of simulation time. If a trajectory fails to rise or settle within this time, then the corresponding characteristic is taken to be 5 seconds.

TABLE I

DESCRIPTION OF TIME-DOMAIN CHARACTERISTICS USED IN THE META-COST, FOR A TRAJECTORY STARTING FROM ITS INITIAL CONDITION RELATIVE TO A FINAL VALUE OF 0.

Characteristic	Description
$\mathtt{RT}_1$	10% to 90% rise time (seconds) of p_{im}
$\mathtt{ST}_1$	2% settling time (seconds) of p_{im}
$\mathtt{OS}_1$	Overshoot (proportion) of p_{im}
$\mathtt{US}_1$	Undershoot (proportion) of p_{im}
$\mathtt{RT}_2$	10% to 90% rise time (seconds) of y_{EGR}
$\mathtt{ST}_2$	2% settling time (seconds) of y_{EGR}
$\mathtt{OS}_2$	Overshoot (proportion) of y_{EGR}
$\mathtt{US}_2$	Undershoot (proportion) of y_{EGR}

A. Results

In the numerical experiments, we perform multiple optimisation trials for the meta-cost tuning of diesel engine control. In half of the trials, we use the re-parametrisation method presented in this paper (referred to as the ‘angular’ parametrisation). As a point of comparison, we use in the other half of the trials a parametrisation where W_Q and W_R are still parametrised as before, however D_Q and D_R are now parametrised simply by their diagonal elements (as positive real numbers). This gives rise to an optimisation problem over an unbounded search space with a 16-length vector decision variable and we refer to this parametrisation as the ‘unbounded’ parametrisation.

We test both parametrisations using each algorithm mentioned in Section IV. For SA, PSO and GA, implementations are available from MATLAB’s Optimization Toolbox. A routine for CMA-ES is available from [30]. These implementations come pre-packaged with bound-constraint handling capability. Algorithm settings (with the exception of termination conditions) are left at their default. Code for implementing NGF was written from scratch, where bound-constraints are handled through projection.

Each optimisation trial is initialised from the value which yield the identity matrices (or a positive scaling thereof) for Q and R (except when using algorithms PSO and GA, which did not require initial guesses). A budget of $20,000$ function evaluations is assigned to each optimisation trial, and all other termination conditions are suppressed. Each trial returns the best meta-cost function value encountered along the run. However, note that since some algorithms (namely CMA-ES, PSO and GA) are population-based, each trial may terminate with slightly in excess of $20,000$ function evaluations. As all these algorithms operate on randomness, an average of 50 trials are taken and reported alongside the sample standard deviation. A table of the results is reported in Table II.

Figure 3 also plots the comparison of the output trajectories[1] between the initial $\{Q, P, R\}$ (with meta-cost value of 0.9303) and the tuned $\{Q, P, R\}$ from the best trial in the sample of CMA-ES with angular parametrisation (which had meta-cost value of 0.2802). This depicts the overall improvement in the

[1]The raw time-series data is commercially sensitive and therefore not included.

TABLE II
RESULTS OF OPTIMISATION TRIALS (50 RUNS PER ROW, EACH RUN WITH 20,000 FUNCTION EVALUATION BUDGET).

Algorithm	Parametrisation	Mean	Std. Dev.
CMA-ES	Angular	0.3235	0.0357
CMA-ES	Unbounded	0.3213	0.0295
NGF	Angular	**0.3812**	0.0874
NGF	Unbounded	0.5523	0.0295
SA	Angular	**0.4542**	0.0313
SA	Unbounded	0.5872	0.0430
PSO	Angular	**0.3432**	0.0603
PSO	Unbounded	0.3673	0.0441
GA	Angular	**0.3721**	0.0531
GA	Unbounded	0.5305	0.0570

controller with the tuning approach, where the tuned controller has visibly reduced the amount of overshoot in both outputs.

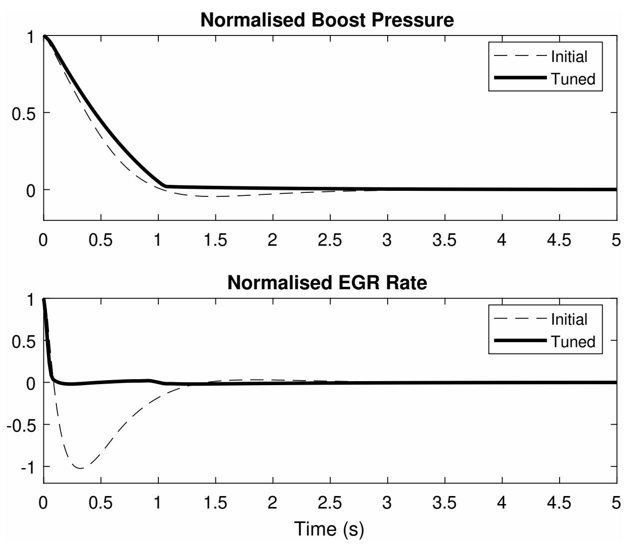

Fig. 3. Normalised output responses of initial (meta-cost: 0.9303) and tuned (meta-cost: 0.2802) controllers.

For each algorithm, we compare the sample of trials obtained from the angular parametrisation against the sample of trials obtained from the unbounded parametrisation. The Wilcoxon-Mann-Whitney rank-sum test [31] is performed to test the null hypothesis that the distribution of trials is equal, against the alternative that a trial using the angular parametrisation is stochastically smaller. We test at the 5% level of significance and list the results in Table III.

TABLE III
WILCOXON-MANN-WHITNEY TEST RESULTS FOR THE DIFFERENCE IN PARAMETRISATIONS.

Algorithm	p-value	Statistically significant?
CMA-ES	0.5261	No
NGF	5.754×10^{-13}	**Yes**
SA	1.070×10^{-16}	**Yes**
PSO	0.0102	**Yes**
GA	1.79×10^{-16}	**Yes**

B. Discussion

Table III shows that there was a statistically significant advantage in using the angular parametrisation for algorithms NGF, SA, PSO and GA. In tandem with the suggestions already provided in Section IV-A, we suggest an informal explanation for this phenomenon. For this particular case study, local minima may have been ‘more dispersed’ over an unbounded search space compared to a box-constrained search space. Compounded with the slight difference in search dimension, an algorithm may have been more efficient at finding local minima with the angular parametrisation in a given budget of function evaluations.

As for CMA-ES, we observe that there was no statistically significant difference from choosing the angular parametrisation over the unbounded parametrisation. Resuming discussion from Section IV, a further property of IGO flow is that of ‘invariance’. Loosely speaking, an algorithm with invariance generalises results on a single problem to a larger class of problems. For example, as CMA-ES operates based on the ordinal ranking of fitness rather than absolute fitness values, it is invariant to strictly positive monotonic transformations of the objective function. Under certain conditions, IGO flow is also invariant to a re-parametrisation of the search space [21]. However, this result does not include the case where the re-parametrisation changes the search dimension and furthermore, the formal guarantee of invariance does not carry over when obtaining implementable discrete-time and finite-sample algorithms from IGO flow (the only takeaway is that CMA-ES is ‘more invariant’ than other algorithms). Hence it is an open question how algorithm performance is affected by a re-parametrisation which changes the search dimension. The empirical results in this paper support the notion that invariance features of CMA-ES may also transfer to such a scenario.

Although is not the main scope of this paper to compare algorithms (since this requires consideration of a wider class of objective function landscapes), we briefly comment that in this particular case study, CMA-ES also achieved the lowest average meta-cost for both angular and unbounded parametrisations, tested to be statistically significant against the other algorithms with the exception of PSO on angular parametrisation (p-values reported in Table IV).

TABLE IV
WILCOXON-MANN-WHITNEY TEST RESULTS FOR THE DIFFERENCE IN ALGORITHM COMPARED TO CMA-ES.

Algorithm	Parametrisation	p-value	Statistically significant?
NGF	Angular	3.5×10^{-5}	Yes
SA	Angular	1.2×10^{-17}	Yes
PSO	Angular	0.0715	No
GA	Angular	2.0×10^{-6}	Yes
NGF	Unbounded	6.4×10^{-18}	Yes
SA	Unbounded	3.5×10^{-18}	Yes
PSO	Unbounded	3.6×10^{-8}	Yes
GA	Unbounded	3.5×10^{-18}	Yes

VII. CONCLUSIONS

The proposed parametrisation method was found to improve optimisation performance on NGF, SA, PSO and GA, while there appeared to be no significant disadvantage when used with CMA-ES. This suggests some merit in choosing the proposed parametrisation in problems of a similar nature regarding the tuning of MPC cost matrices with a meta-cost function. Moreover, the box-constrained parametrisation alleviates any numerical issues stemming from the decision variables growing unbounded. The results presented warrant further investigation (theoretical and applied) into establishing the benefits of this re-parametrisation, as well as studying the invariance properties of CMA-ES when the re-parametrisation involves a change in the search dimension. The work could also be extended to using a nonlinear diesel engine model in simulation for offline tuning, and implementing the tuned MPC on a physical diesel engine.

REFERENCES

[1] J. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[2] G. S. Sankar, R. C. Shekhar, C. Manzie, and H. Nakada, “Efficient calibration of real-time model-based controllers for diesel engines — part II: Incorporating practical robustness guarantees,” in *IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017.

[3] R. C. Shekhar, G. S. Sankar, C. Manzie, and H. Nakada, “Efficient calibration of real-time model-based controllers for diesel engines — part i: Approach and drive cycle results,” in *IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017.

[4] R. Soeterboek, *Predictive Control: A Unified Approach*. Prentice Hall, 1992.

[5] R. Shridhar and D. J. Cooper, “A tuning strategy for unconstrained multivariable model predictive control,” *Industrial & Engineering Chemistry Research*, vol. 37, no. 10, pp. 4003–4016, 1998.

[6] G. S. Sankar, W. H. Moase, R. C. Shekhar, T. J. Broomhead, and C. Manzie, “Towards systematic design of MPC to achieve time domain specifications,” in *Australian Control Conference*, 2015.

[7] A. S. Ira, I. Shames, C. Manzie, R. Chin, D. Nesic, H. Nakada, and T. Sano, “A machine learning approach for tuning model predictive controllers,” in *International Conference on Control, Automation, Robotics and Vision*, 2018.

[8] R. Susuki, F. Kawai, C. Nakazawa, T. Matsui, and E. Aiyoshi, “Parameter optimization of model predictive control using PSO,” in *SICE Annual Conference*. IEEE, Aug. 2008.

[9] H. Waschl, D. Alberer, and L. del Re, “Numerically efficient self tuning strategies for MPC of integral gas engines,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 2482–2487, jan 2011.

[10] G. Gous and P. de Vaal, “Using MV overshoot as a tuning metric in choosing DMC move suppression values,” *ISA Transactions*, vol. 51, no. 5, pp. 657–664, sep 2012.

[11] R. Shepard, S. R. Brozell, and G. Gidofalvi, “The representation and parametrization of orthogonal matrices,” *The Journal of Physical Chemistry A*, vol. 119, no. 28, pp. 7924–7939, 2015.

[12] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*. Addison Wesley, 2000.

[13] N. Hansen, “The CMA evolution strategy: A tutorial,” University of Paris-Saclay, Tech. Rep., 2016.

[14] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *ICNN International Conference on Neural Networks*. IEEE, 1995.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992.

[16] Y. Nesterov and V. Spokoiny, “Random gradient-free minimization of convex functions,” *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2015.

[17] M. L. Derouiche, S. Bouallègue, J. Haggège, and G. Sandou, “LabVIEW perturbed particle swarm optimization based approach for model predictive control tuning,” *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 353–358, 2016.

[18] J. van der Lee, W. Svrcek, and B. Young, “A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making,” *ISA Transactions*, vol. 47, no. 1, pp. 53–59, Jan. 2008.

[19] L. Ingber, “Adaptive simulated annealing (ASA): Lessons learned,” *Control and Cybernetics*, vol. 25, no. 1, pp. 33–54, 1996.

[20] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi, “Theoretical foundation for CMA-ES from information geometry perspective,” *Algorithmica*, vol. 64, no. 4, pp. 698–716, 2011.

[21] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen, “Information-geometric optimization algorithms: A unifying picture via invariance principles,” *Journal of Machine Learning Research*, vol. 18, no. 18, pp. 1–65, 2017.

[22] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík, “Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009,” in *Genetic and Evolutionary Computation Conference*, 2010.

[23] A. Auger, N. Hansen, J. P. Zerpa, R. Ros, and M. Schoenauer, “Experimental comparisons of derivative free optimization algorithms,” in *8th International Symposium on Experimental Algorithms*, 2009.

[24] J. Villemonteix, E. Vazquez, M. Sidorkiewicz, and E. Walter, “Global optimization of expensive-to-evaluate functions: an empirical comparison of two sampling criteria,” *Journal of Global Optimization*, vol. 43, no. 2-3, pp. 373–389, 2008.

[25] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[26] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*, 5th ed. Nob Hill Publishing, 2009.

[27] E. Gilbert and K. Tan, “Linear systems with state and control constraints: the theory and application of maximal output admissible sets,” *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[28] J. Wahlstrom and L. Eriksson, “Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 225, no. 7, pp. 960–986, jul 2011.

[29] C. Schmid and L. Biegler, “Quadratic programming methods for reduced hessian SQP,” *Computers & Chemical Engineering*, vol. 18, no. 9, pp. 817–832, 1994.

[30] N. Hansen, “CMA-ES source code.” [Online]. Available: http://cma.gforge.inria.fr/cmaes_sourcecode_page.html

[31] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*, 3rd ed. Wiley, 2014.

APPENDIX A
CLOSED-LOOP STABILITY

We illustrate the concept of closed-loop stability with a simple example. Consider the linear dynamical system $x^+ = Ax + Bu$ with a full-state feedback control law $u = \kappa(x)$. Then the closed-loop stability refers to the stability of the dynamical system $x^+ = Ax + B\kappa(x)$. Suppose a one-dimensional example where $x \in \mathbb{R}$, $u \in \mathbb{R}$ with $A = 1.2$, $B = 1$ and a linear control law $\kappa(x) = -0.1x$. The closed-loop system is then $x^+ = 1.1x$, which is unstable at the origin because $\|x\| \to \infty$ as time $k \to \infty$, for all initial conditions $x_0 \neq 0$.

Now suppose $\kappa(x) = -0.3x$. Then the closed-loop system $x^+ = 0.9x$ is stable at the origin because $\|x\| \to 0$ as time $k \to \infty$, for any initial condition. The results found in Section V and within [25] extend closed-loop stability guarantees for linear constrained systems to more complicated feedback control laws of forms similar to (5).